



## HOW TO GET A WIN32 HBITMAP TO WINRT/UWP SOFTWAREBITMAP

In the era of machine learning and visual recognition it goes without saying that passing graphical information in a more common scenario. This code snippet hopes to help the many queries that try to accomplish it in the Windows ecosystem. Especially going from Win32 to WinRT/UWP world.

```
using Windows.Media.Capture;
using Windows.Storage.Streams;
using Windows.UI.Xaml.Media.Imaging;

// Capture picture info and plug the HBITMAP
IPicture picture; // Holds our picture data
IRandomAccessStream stream; // Staging bridge buffer to WinRT/UWP

// Buffer receiving bitmap data.
IRandomAccessStream;
```

Obtain bitmap buffer from COM API

```
bool result = false;

// Create a PICTUREDESC structure.
PICTDESC picDesc;
picDesc.Type = PICTYPE_BITMAP;
picDesc.hBitmap = hBitmap;
picDesc.hPal = NULL;

// Create an IPicture object.
HRESULT res = OleCreatePictureIndirect(&picDesc, IID_IPicture, false,
    reinterpret_cast<void*>(&picture));

if (!SUCCEEDED(res))
    return false;

// Create a staging buffer.
IRandomAccessStream = CreateStreamOnHGlobal(0, true, &stream);

if (!SUCCEEDED(res))
    return false;

// Copy the bitmap data to the staging buffer.
picture->Release();
return true;
```

Obtain bitmap buffer from COM API world to WinRT/UWP space.

```
IRandomAccessStreamOverStream(stream, BSOS_DEFAULT, IID_PPV_ARGS(&spRandomAccessStream))
```

Once we have the RandomAccessStream populated it's a vanilla WinRT walk in the park:

```
using Windows.UI.Xaml.Media.Imaging;

// Create a BitmapDecoder object.
BitmapDecoder decoder = new BitmapDecoder(spRandomAccessStream.Get());

// Get the software bitmap.
SoftwareBitmap softwareBitmap = decoder.GetSoftwareBitmapAsync().GetAwaiter().GetResult();

// softwareBitmap is your bitmap in WinRT/UWP World!
```



});

wait for the continuation to finish and handle any error that occurs.

```
wcout <<< L"Waiting for tasks to finish..." <<< endl;
continuation.wait();

// Alternatively, call get() to produce the same result.
//continuation.get();

try {
    // ...
} catch (const exception& e) {
    wcout <<< L"Caught exception." <<< endl;
}
```



Multiplayer Procedural Challenges & Involving Feature Spaces

Powers Of The Dot Product

